



Arquitectura de las Computadoras

Informe TPE

Integrantes

- Tarradellas del Campo, Manuel Legajo 58091
- Karpovich, Lucia Legajo 58131
- Rolandelli, Alejandro Legajo 56644

2° Cuatrimestre - 2018

Para este proyecto nos adaptamos a una separación y diseño similar al aplicado en el “BareBones”. Podemos simplificar dicha separación mencionando *Userland* y *Kernel*. Estos dos directorios representan diferentes secciones de memoria del sistema, separadas para una mayor seguridad, ya que el Kernel es el único capaz de comunicarse con el hardware del sistema. Esto lo logra a través de la implementación de Drivers con las funciones necesarias para el manejo de las interrupciones de dicho hardware. Si el usuario desea utilizar cierto hardware debe usar Módulos con funciones que se comunican con el Kernel a través de System Calls, otorgadas por el Kernel, y este último es el que decide finalmente como actuar frente a los pedidos del Userland.

Fue desde el inicio que se planteó que sería de este modo ya que priorizábamos la seguridad y correcta separación, y este era un claro pro de este formato. Fue con el desarrollo del trabajo que fueron apareciendo los contras más obvios. Entre ellos se destacaron la repetición de ciertos códigos y el hecho de que requirió módulos y funciones que actuaran como comunicadores entre User y Kernel, esto nos forzó a tener un código más complejo de seguir dado que cada acción pasa por una serie de archivos y funciones para poder cumplir su propósito.

El TPE está separado de forma tal que se agrupan las funciones según su objetivo y área de acción. Esto nos dejó como resultado varios grupos, entre los principales se encuentran: *Video*, *Time*, *Keyboard* y *Sound*. Estos son llamados desde *Userland* a través de la Shell, la cual hace uso de los modulos “comunicadores” respectivos, los cuales a su vez hacen la correspondiente llamada, a través de System Calls, a los drivers que se encuentran en el *Kernel*.

A lo largo del desarrollo del proyecto se nos presentaron una serie de problemas. El primero fue hacer el manejo del modo video. Esta situación requirió mucha investigación dado el poco conocimiento que teníamos sobre el tema; pero más allá de esto, fue un desafío dado que estábamos acostumbrados a hacer uso de funciones para imprimir ‘strings’ sin darle mucha consideración. El modo video hizo

que tuviéramos que crear dichas funciones partiendo de dibujar pixel a pixel en la pantalla.

Fueron apareciendo problemas menores los cuales mencionaremos brevemente ya que su solución no presento grandes dificultades. En un principio usamos una estructura para el modo video que no funcionaba para lo que necesitábamos pero cuando llegamos a esta conclusión fue simplemente buscar otra. También este modo presento dificultades a la hora de buscar la localización de la estructura y de su activación. El beep no fue complejo de realizar ya que se encontró información sencilla de entender de manera rápida, lo que no esperábamos era su activación a través de un flag en el archivo run. Esto genero una cascada de cambios innecesarios e investigación para encontrar y solucionar lo que no estaba funcionando.

El manejo de Syscalls y Excepciones no resultó dificultoso, dado que era mucho de lo que se nos había enseñado en la teoría, y el necesario manejo de Assembler requerido para completarlo no era mucho más complejo que lo visto en la práctica de la primera parte de la materia.

Por último, algo que no logramos resolver fue liberar el stack de la función que sufrió una excepción ya que al llamar a la rutina de atención de la excepción antes de liberarse, queda en memoria lo cual puede acabar la memoria disponible si sucede demasiadas veces. Esto se pensó resolver liberando el stack de dicha función en el código Assembler de la interrupción, utilizando el registro RBP que tiene el valor de la base de su stack. El problema no se pudo resolver ya que no estábamos seguros de que valor poner en el registro RSP para liberar el stack y luego retornar correctamente a la consola.

Para concluir, el trabajo incluyo conocimiento y manejo de cosas que no conocíamos, por lo cual, una gran parte de este consistió en investigación detallada. Como resultado de esto aprendimos mucho más de lo que esperábamos al inicio del proyecto, no solo sobre cosas vistas en la materia, las cuales se profundizaron; sino que también cosas nuevas como las mencionadas anteriormente (sonido, modo video, etc).